Research Heaven,
West Virginia

World’s most civil servant

World’s worst title

Martha: a next generation testable language

Not a “language” but AI agents to help analysts explore a range of model behaviors

Tim Menzies
West Virginia University
tim@timmenzies.net

Problem

- **The unknown knowns**
  - Factors that have been modeled separately,
  - but never studied in conjunction

		us	
		What we know	What we don't know
them	What they know	The known knowns	The unknown knowns
	What they don't know		The unknown unknowns

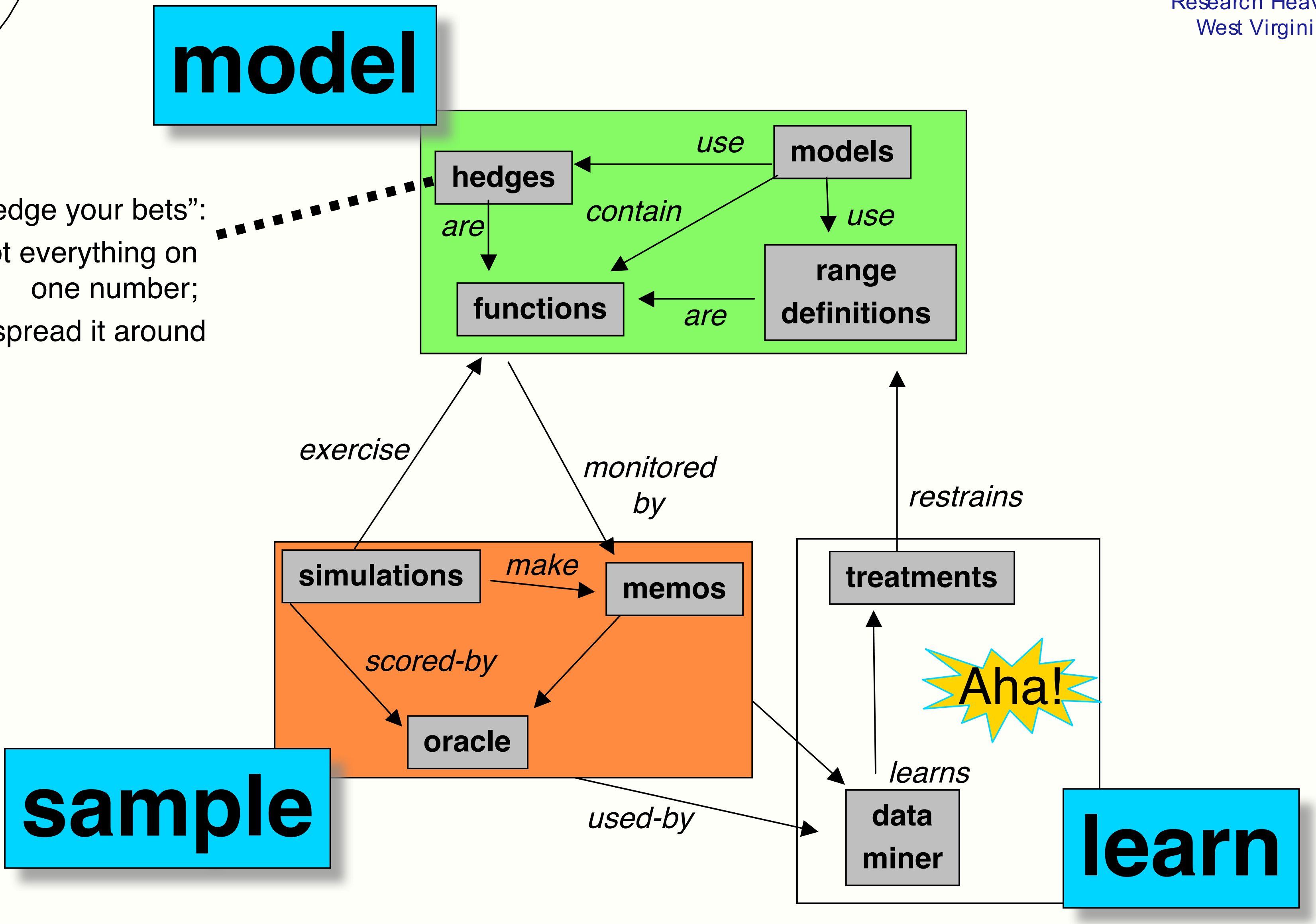
Leveson:

- common cause of software failure
- components that are known to fine in isolation but failing when combined due to an unknown interaction.

Brian O'Conner:

- Columbia incident- pre-launch foam-strike studies
- Did not study a critical combination of factors

Approach



model

hedges

models

use

contain

use

are

functions

are

range definitions

“Hedge your bets”:
Not everything on one number;
spread it around

exercise

monitored by

restrains

simulations

make

memos

scored-by

oracle

sample

treatments

Aha!

learns

data miner

used-by

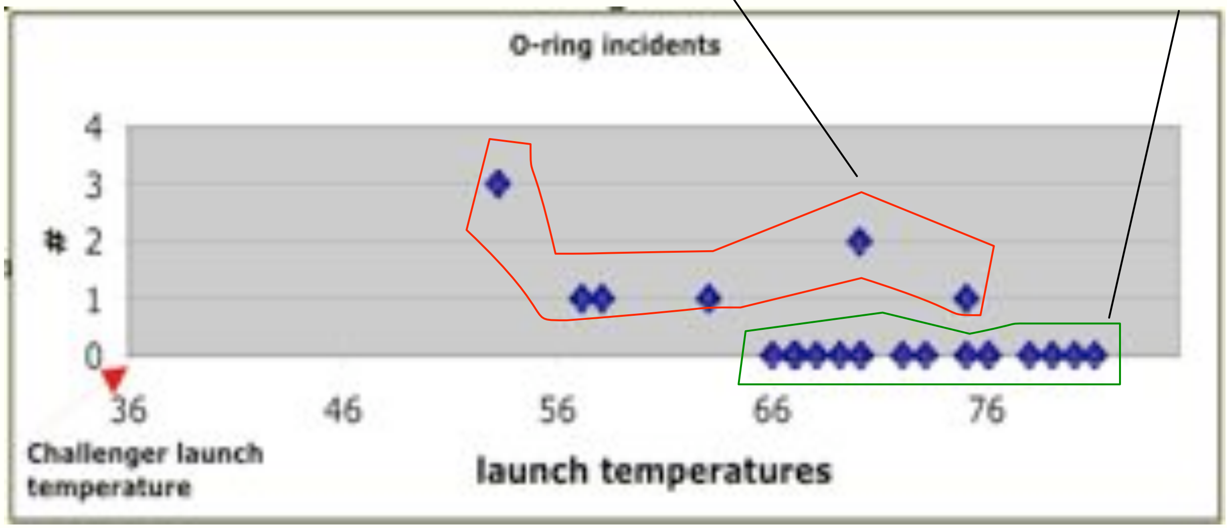
learn

Importance/ Benefits

The trouble isn't what people don't know;
it's what they do know that isn't so. -- Will Rogers

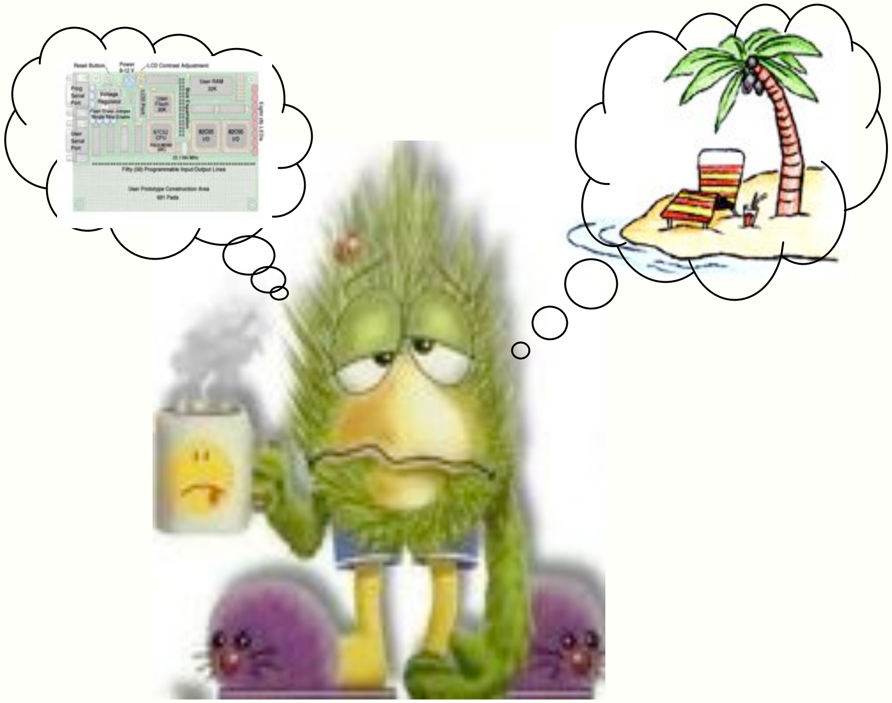
Q1: Cold causes o-ring incidents?
A1: No, so launch

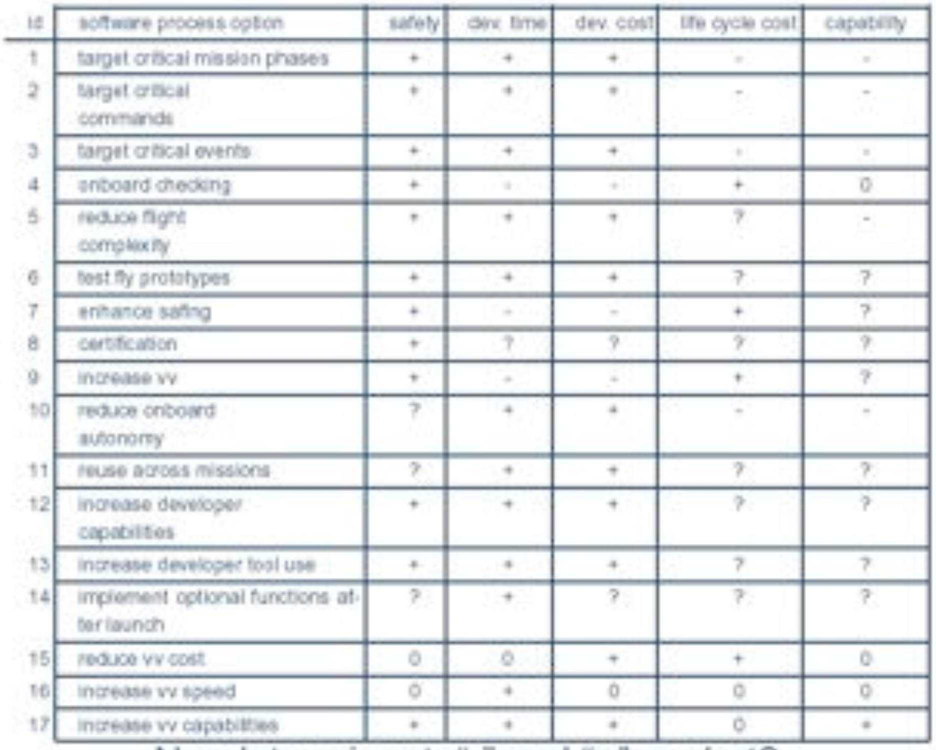
Q2: Heat blocks o-ring incidents?
A2: Yes, don't launch



My ideal computer does not say "hello world"; rather it says "hello, that's strange". -- Tim Menzies

Relevance To NASA

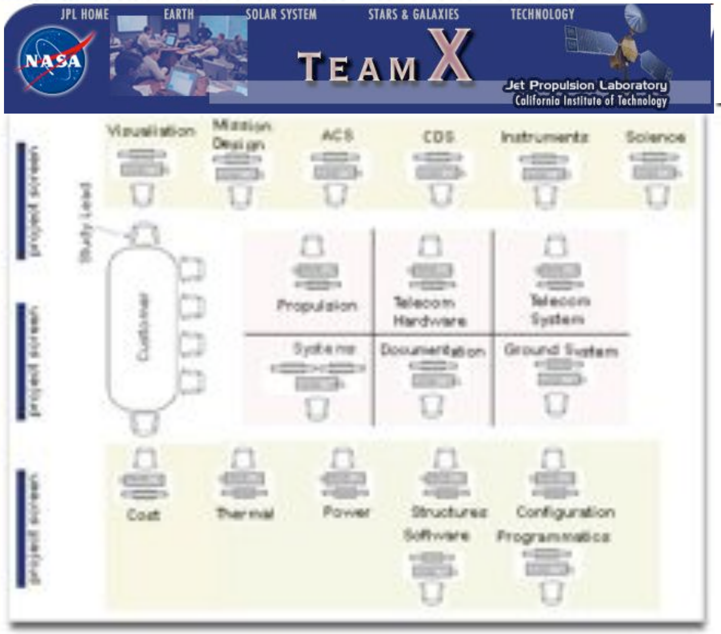




id	software process option	safety	dev. time	dev. cost	life cycle cost	capability
1	target critical mission phases	+	+	+	-	-
2	target critical commands	+	+	+	-	-
3	target critical events	+	+	+	-	-
4	onboard checking	+	-	-	+	0
5	reduce flight complexity	+	+	+	?	-
6	test fly prototypes	+	+	+	?	?
7	enhance safing	+	-	-	+	?
8	certification	+	?	?	?	?
9	increase vv	+	-	-	+	?
10	reduce onboard autonomy	?	+	+	-	-
11	reuse across missions	?	+	+	?	?
12	increase developer capabilities	+	+	+	?	?
13	increase developer tool use	+	+	+	?	?
14	implement optional functions after launch	?	+	?	?	?
15	reduce vv cost	0	0	+	+	0
16	increase vv speed	0	+	0	0	0
17	increase vv capabilities	+	+	+	0	+

Nondeterminant: "-" and "+" = what?

- **Other case studies**
  - SILAP: Error potential calculations for NASA projects
  - NEAR: Near Earth Orbit Rendezvous
  - Team X: rapid development mission concepts
  - Learning controllers for sounding rockets

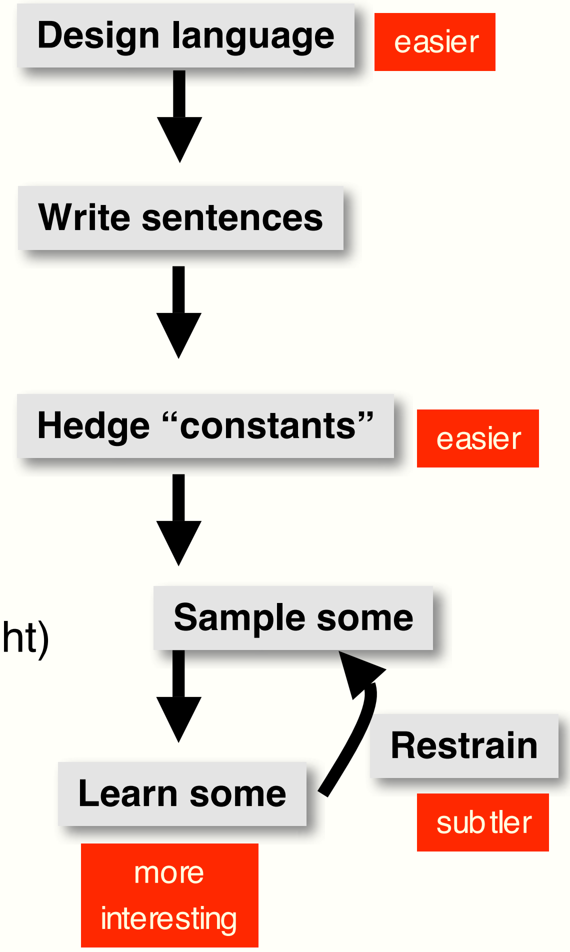


- **How to audit software process knowledge built by a team?**
  - How ensure that, in the heat of the moment, critical features of a mission are not over-looked?
  - Or worse, accidently over-written by other decisions?

Accomplishments

Research Heaven,
West Virginia

- **Before:**
  - –Three (non-NASA) case studies
  - –Lessons:
    - Surprisingly fast learning of controllers
    - Automating a “principle-investigator-in-a-box”
- **Now:**
  - –Recognition of three problems
    - The learning problem (more interesting that we’d thought)
      - SURFER: generalized iterative learning environment
      - Case study: JPL DDP/ Team X
    - The modeling problem (easier than we’d thought)
      - Case study: SILAP (IV&V model of project error potential)
      - Case study: near earth orbit autonomous rendezvous
      - Case study: control options, sounding rocket
    - The restraining problem (more subtle that we’d thought)
      - Exploration vs exploitation



Next Steps

- **More case studies**
  - SILAP: lots to do
  - Team X: excellent test bed
  - Synergy with HRT project on cost-benefits autonomous systems
- **Generalization**
  - N case studies
  - Reusable “marthas” extracted from the case studies
- **Better restraining policies**
  - Use internals of data miner to define what to try next
    - Bayesian analysis